

ModelSim/ModelSim-Altera

入力波形(テストベンチ)作成機能を用いた シミュレーション

ver.10



2013年2月 rev.1

ELSENA,Inc.





入力波形(テストベンチ)作成機能を用いたシミュレーション

<u>目次</u>

1.	はじめに	
2.	操作方法	
	2-1. 新規プロジェクトの作成	
	2-2. ファイルの作成と登録	
	2-2-1. 新規デザイン・ファイルの作成5	
	2-2-2. 既存デザイン・ファイルの登録7	
	2-3. コンパイル実行順番の指定	
	2-4. デザイン・ファイルのコンパイル	
	2-5. 入力波形の作成	
	2-5-1. Wave ウィンドウヘピンを登録8	
	2-5-2. Create Pattern Wizard を用いて入力波形を作成9	
	2-5-3. 波形の編集	
	2-6. テストベンチの生成	
	2-7. テストベンチの追加登録とコンパイル 15	
	2-7-1. テストベンチの追加15	
	2-7-2. テストベンチのコンパイル	
	2-8. デザインのロード	
	2-9. シミュレーションの 実行	
	2-10. シミュレーションとプロジェクトの終了	

ELSENA



EISENA

1. <u>はじめに</u>

Quartus[®] II で FPGA および CPLD の開発を進めるときに、適所で ModelSim[®] または ModelSim-Altera[®] を用いて、論理合成前の "ファンクション・シミュレーション (RTL シミュレーション)" と、 配置配線後の "タイミング・シミュレーション" を行い、動作検証します。

その際、ModelSim および ModelSim-Altera には、入力波形をグラフィカルに作成する機能があり、さら に、作成した入力波形を HDL 言語に変換してエクスポート(生成)することができます。容易にテストベンチ を作成することで容易にシミュレーションが行えます。

本資料では、ModelSim および ModelSim-Altera のファンクション・シミュレーションにおいて、入力波形 作成機能を用いてテストベンチを作成し、シミュレーションを行うフローをご紹介します。

- ※ ModelSim-Altera においてこの機能を利用する場合は、ver. 6.6c 以降をご利用ください。
- ※ 本資料では、ModelSim-Altera の画面を用いて説明しています。
- ※ ModelSim-Altera の基本的なシミュレーション方法に関して、下記資料も併せてご参考ください。これらは、本資料を ご入手になった技術情報サイトにてご入手いただけます。

資料タイトル 『ModelSim-Altera Edition RTL シミュレーション - VHDL/VerilogHDL - 』 『ModelSim-Altera Edition タイミング・シミュレーション - VHDL/VerilogHDL - 』

対象バージョン: Quartus II v12.1 SP1

ModelSim-Altera 10.1b ModelSim 10.1d

2. 操作方法

入力波形作成機能を用いたファンクション・シミュレーションの操作フローは以下のとおりです。

- 2-1. 新規プロジェクトの作成
- 2-2. ファイルの作成と登録
- 2-3. コンパイル実行順番の指定 ・・・VHDL ユーザのみ実施
- 2-4. デザイン・ファイルのコンパイル
- 2-5. 入力波形の作成
- 2-6. テストベンチの生成
- 2-7. テストベンチの追加登録とコンパイル
- 2-8. デザインのロード
- 2-9. シミュレーションの実行
- 2-10. シミュレーションとプロジェクトの終了

2-1. 新規プロジェクトの作成

ModelSim および ModelSim-Altera を起動し、シミュレーション用の新規プロジェクトを作成します。既に プロジェクトを作成している場合はこの操作は省略し、次項 2-2.ファイルの作成と登録をご覧ください。

- ② プロジェクト名、プロジェクト・フォルダを設定します。Create Project ウィンドウにおいて、以下の項目を 指定し、OK ボタンをクリックします。
 - Project Name ・・・・・ プロジェクト名を入力
 - Project Location ····· プロジェクトを作成する場所(フォルダ)を指定
 - Default Library Name ···· ライブラリ・フォルダ名を入力
 - ※ Project Location で指定した場所に、Default Library Name で指定した名前 (work) のフォルダとプロジェク ト・ファイル (.mpf) が自動的に生成されます。

<u>F</u> ile <u></u>	<u>E</u> dit	<u>V</u> iew	<u>C</u> ompile	<u>S</u> imulate	A <u>d</u> d	L <u>i</u> brary	
Nev	v			Eolde	r	<i>4</i> 4	
<u>O</u> pe	en			Sourc	e	, 	🕅 Create Project 🛛 🛛
<u>L</u> oa	d			▶ <u>P</u> roje	ct		Project Name
⊆los	se			Librar	у		filtref
I <u>m</u> p	ort			▶ D <u>e</u> bu	g Archiv	/e [EL_1	Project Location
<u>E</u> ×p	ort			Lib	rary	\$MODEL_1	work/adesigns/fir_filter/func_sim_Browse
Carr			Chilas	I Lih	rarv	\$MODEL 1	Default Library Name work Copy Settings From I.1/modelsim_ae/modelsim.ini Browse Copy Library Mappings OK

以上でプロジェクトの作成は完了です。

OK ボタンをクリック後、Add items to the Project ウィンドウが表示されます。このウィンドウを活用して 次の操作(デザインの登録・作成)を行うことも可能ですが、この資料では、メニューから操作する方法を ご案内します。そのため、Add items to the Project ウィンドウを Close ボタンで閉じます。

M Add items to the	Project 🛛 🔀
Click on the icon to a	add items of that type:
Create New File	Add Existing File
Create Simulation	Create New Folder
	Close

※ 本資料は、VHDL デザインにおける操作をサンプル画像に用いてご紹介しています。VerilogHDL ユーザの方は、一 部操作および画像が異なる箇所がございますのでご注意ください。



【補足①:アルテラ・ライブラリ】

ALTIMA

ModelSim-Altera の場合、作成したプロジェクト・ファイルにはコンパイル済みの各種アルテラ・ライブラリ(メガファンクショ ン・ライブラリ、デバイス・ライブラリなど)のパス情報が記録されます。下図は mpf ファイルの記述(一部)サンプルです。 mpf ファイルはテキスト・エディタで開くことが可能です。このようにアルテラ・ライブラリへのパス情報が自動的に記録される ため、アルテラ専用のメガファンクション・ライブラリやデバイス・ライブラリの準備作業を行う必要がありません。正規版の ModelSim (PE/SE など)の場合には、コンパイル済みのアルテラ・ライブラリが用意されていないため、ライブラリの作成や 指定作業(ライブラリのマッピング)を行う必要があります。ご注意ください。



2-2. ファイルの作成と登録

作成したプロジェクトにおいて、シミュレーション実行に必要なデザイン・ファイルを作成および登録します。

2-2-1. 新規デザイン・ファイルの作成

新規でデザイン・ファイルを作成します。既にデザイン・ファイルがある場合にはこの操作は省略し、次項 2-2-2.既存デザイン・ファイルの登録をご覧ください。

- ① Project ウィンドウを選択し、アクティブにします。
- ② Project \checkmark ニュー \Rightarrow Add to Project \Rightarrow New File... を選択します。
- ③ 表示された Create Project File ウィンドウ内の各項目を入力し、OK ボタンをクリックします。

M Create Project File		
File Name		
filtref	Bro	owse
Add file as type VHDL Verilog SystemVerilog VHDL PSL	Folder Top Level OK	Cancel
Text SystemC	File Name	ファイル名を入力
TCL Macro	Add file as type	ファイルのフォーマットをプルダウンリストから選択
↓	Folder	ファイルの保存先を指定

※ File メニュー \Rightarrow New \Rightarrow Source から新規ファイルを生成することもできますが、その場合作成したファイルは<u>自動</u> 的にプロジェクトへ登録されません。ご注意ください。



EISEN

④ Project ウィンドウに作成したファイルが登録されていることを確認します。

Project - F(/dosign	<pre>work/qdesign</pre>	s/fir_filter/func_sim/fil	tref 🚃
Name	△ Status	Туре	
💼 filtref.vhd	\$	VHDL	
👖 Library 🗙 🛗 Pro	oject 🗙		

- ⑤ Project ウィンドウに登録されたファイルを選択し、右クリック ⇒ Edit(またはファイルをダブルクリック) で編集します。
- ⑥ 編集したファイルを保存します。File ⇒ Save または、ツールバーの 🔚 アイコンをクリックします。
 - ※ コンパイル時に自動保存はされません。必ずコンパイルの前にファイルを保存してください。

【補足②:テキスト・エディタの指定】

デザイン・ファイルを編集する際に好みのテキスト・エディタを使用する場合は、以下の手順で設定可能です。

- 1. Tools メニュー \Rightarrow Edit Preferences \Rightarrow By Name タブ \Rightarrow Main グループの"+"マークをクリックにて展開 し、Preference Item より Editor をダブルクリックで選択します。
- 2. Change Main Preference Value ダイアログボックスに、使用するテキスト・エディタの実行プログラムの絶対 パスを記述し、OK ボタンですべてのダイアログボックスを閉じます。

	Preferences				
	Preferences:				
I	Preference Item	Value	Description		
	— DefaultVirtualDatas	virtuals	for viewing log	files during an active s	
I	— DefaultVirtualFileNa	virtuals.do	the default filename used when savi		
	— DisplayDatasetPrefi	1	turns dataset prefix viewing on or of		
	- EchoBatchCommand	<u>0</u>	If set to 1, cor	mmands entered intera	
I	Editor	M Change Main Preference Va	lue 🛛 🛛		
	— EnableCommandHel			bles command comple	
I	- EnableFileHelp	New value for		bles file pathname co	
I	— errorColor	"PrefMain(Editor)":		or messages displaye	
		ここへ*.exe の絶対パスを入力	する。	successful Project con	
I	- failureColor		Capital	f failures that are rer	
I	EEMDelay			cus takes effect whe	
I	file	trapscript	name of the fil	e for caving transcript	
	Editor – EnableCommandHel – EnableFileHelp – errorColor – errorProjectCompile – failureColor – FFMDelay – file	M Change Main Preference Va New value for "PrefMain(Editor)": <i>ここへ*.exe の絶対パスを入力</i> Ok transcript	lue 🛛	bles command comp bles file pathname c or messages display auccessful Project co f failures that are re rous takes effect wh e for saving transcrip	

- 3. File メニュー⇒ Open またはショートカットアイコン ²²より、編集するデザイン・ファイルを開きます。
- 4. 指定したファイルが ModelSim 内蔵のテキスト・エディタ(Source ウィンドウ)にて開きます。
- 5. Source ウィンドウを選択しアクティブにした状態のまま、Source メニュー ⇒ Open in external editor を選択 します。(その他のウィンドウが選択されている場合、Source メニューは表示されません。)
- 6. 好みのテキスト・エディタで編集が行えます。

ファイルを編集後、保存し閉じると、下図のワーニング・メッセージが表示されるので、Reload を選択してください。 Source ウィンドウで開いていたデザイン・ファイルに保存内容が反映されます。

M Warni	ng 🛛
	FILE: The Society of Strandweight Management of Strandsociety (Strandsociety)
	This file has been modified outside of the source editor.
	You may choose to:
	1) OVERWRITE disk changes with current editor content 2) RELOAD the changes from disk 3) Ignore the difference
	Overwrite Reload Ignore



2-2-2. 既存デザイン・ファイルの登録

既存のデザイン・ファイルをプロジェクトへ登録します。

- ① Project ウィンドウを選択し、アクティブにします。
- ② Project $\checkmark = \neg \rightarrow$ Add to Project \Rightarrow Existing File... を選択します。

M Add file to Project		×
File Name		
1		Browse
Add file as type	Folder Top Level	_
Reference from current location	Copy to project directory	
	ОК	Cancel

 Add file to Project ウィンドウ内の File Name 脇にある Browse ボタンをクリックし、既存のファイルを 選択後、OK ボタンをクリックします。

デザイン・ファイルが同じフォルダ内に保存されている場合は、同時に複数選択が可能です。保存されているフォルダ が異なる場合には、その都度デザイン・ファイルを指定してください。

④ Project ウィンドウに指定したデザイン・ファイルが登録されていることを確認します。

<u>F</u> ile	<u>E</u> dit	Vie	W	<u>C</u> om	npil	e	<u>S</u> im	ulat	е	A <u>d</u> d		Proj	ect	T <u>o</u> ol	s	Layo <u>u</u> t
	• 🚔		3	Ŵ		ğ.	9		»	\geq		\bigcirc	∘ ĝ¢	8 -	м	
anna Pr	oject -	F:/de	esigr	n_sam	ple	/wo	rk/q	desig	ns/	fir_fil	ter	/func	_sim	/filtref	=	
🔻 Nam	ne				Δ	Sta	atus				Ty	ре				
	acc.v	/hd				?					٧H	DL				
	accu	m.vh	d			2					٧H	DL				
	🖣 filtrel	f.vhd	ł			?					٧H	DL				
	🖣 hvalu	jes.v	hd			?					٧H	DL				
	🖥 mult.	vhd				?					٧H	DL				
	🕯 state	_m.v	/hd			?					٧H	DL				
	👌 taps.	vhd				?					٧H	DL				

2-3. コンパイル実行順番の指定

デザインのシミュレーション用入力波形を作成するにあたり、事前にデザイン・ファイルをコンパイルしま す。その際 VHDL デザインでは、コンパイルの実行順番を事前に指定する必要があります。(VerilogHDL デザインでは不要な操作ですので、VerilogHDL ユーザは、次項 2-4.デザイン・ファイルのコンパイルをご覧 ください。)

下位階層のファイル(パッケージやユーザ・ライブラリなど)から順番にコンパイルし、最後に最上位階層のファイルをコンパイルします。今はまだテストベンチがありませんので、現段階での最上位階層は、FPGA(または CPLD)のトップデザインがそれにあたります。

以下の手順でコンパイルを実行するファイルの順番を設定してくだ さい。

- 1. Compile メニュー ⇒ Compile Order を選択します。
- Compile Order ウィンドウにおいて、ファイルの順番を ▲ または ▼ ボタンにて並び替えます。画面上位が下位階層 ファイル、下に向かうほど上位階層ファイルとなるように並 べ、一番下が最上位階層ファイルです。
- 3. 並び替えが完了したら OK ボタンで Compile Order ウィン ドウを閉じます。

M Compile Order		
Current Order		
acc.vhd acc.vhd m acc.vhd m hvalues.vhd m tult.vhd m taps.vhd m taps.vhd	長下位階層	
nitrer.vnd	長上位階層	
		▲ ₩
		**
	-	″ ¾ ※ »
	Auto Generate OK C	ancel

2-4. デザイン・ファイルのコンパイル

プロジェクトに登録したシミュレーションに必要なファイルをすべてコンパイルします。

◆ 登録したすべてのファイルを<u>一度に</u>コンパイルする場合

Compile メニュー ⇒ Compile All を選択します。

◆ 各ファイルを個別にコンパイルする場合

Project ウィンドウ内のファイルを右クリックで選択し、Compile ⇒ Compile Selected を選択します。

コンパイルが問題なく完了しているか、エラーはないかについては、Transcript ウィンドウに表示されるメッ セージの他に、Project ウィンドウの Status 欄の表示で確認することができます。

もしエラーがあった場合にはその箇所と内容を確認し、記述を修正してください。そして、コンパイルが未 完了のファイルのみを再度コンパイルします。(すべてのファイルを実行する必要はありません。)最後に、す べてのファイルにエラーがないことが確認できれば、コンパイル終了です。

📷 Project - F:/design_	_sample/work/qdesig	ns/fir_filter/func_sim/filtr	ef	
Name	△ Status	Туре	Ord	(
📷 acc.vhd	×	VHDL	1	
📷 accum.vhd		VHDL	0	2
📷 filtref.vhd	?	VHDL	6	3
📷 hvalues.vhd		VHDL	2	10
📷 mult.vhd	?	VHDL	3	
📷 state_m.vhd	?	VHDL	5	
📷 taps.vhd	1	VHDL	4	\sim

?	未コンパイル	
1	コンパイル済(エラーなし)	
×	コンパイル済(エラーあり)	
		Ϊ

2-5. 入力波形の作成

最上位デザインの各ピンに入力する信号を作成します。

2-5-1. Wave ウィンドウヘピンを登録

信号を波形表示する Wave ウィンドウを起動し、デバイスのピンを登録します。

- ① Library タブを選択し、アクティブにします。
- ② 現プロジェクトのワーキング・ライブラリを展開し、最上位デザインを右クリックで指定 ⇒ Create Wave を 選択します。
- ③ Wave ウィンドウが起動し、信号が登録されていることを確認します。







2-5-2. Create Pattern Wizard を用いて入力波形を作成

Wave ウィンドウに登録した入力ピンに対し、信号(波形)を作成します。

① 目的の信号を右クリックで選択し、Edit メニュー ⇒ Create/Modify Waveform を選択します。

💼 Wave - Default 🚃			
*	Msg:	s	
<pre> /filtref/clk×2</pre>	No Data Object Declaration	-	
	Edit View	Cut Copy	Ctrl+X Ctrl+C
	Radix Format	Paste Delete	Ctrl+V
	Combine Signals Group	Create/Modify V Map To Design S	Waveform Signal

- ② Create Pattern Wizard が起動します。
- ③ Select Pattern セクションの Patterns から作成する波形パターンを選択します。
- ④ 波形の Start Time、End Time を入力し、単位を選択後、Next > ボタンをクリックします。

Counter V	Generate a waveform for any signal for the chosen pattern. The allowed patterns are: Constant Clock Random Repeater Counter	Select Pattern Signal N Patterns Signal N Colock NewS Constant Start Tir Random 0 C Repeater C Counter	lame ig:/filtref/clkx2 me End Time 1000	Time Unit
-----------	--	--	--	-----------

- ⑤ 選択した波形パターンごとに異なるウィンドウが立ち上がります。(詳細は後述)
- ⑥ ウィンドウ内の値を入力した後、Finish ボタンをクリックします。
- ⑦ Wave ウィンドウに波形が反映されます。

波形パターンは、Clock(クロック)/Constant(定数)/Random(ランダム値)/Repeater(繰り返し)/Counter(カウント値)の5種類です。画面の設定内容は次のとおりです。

Clock

クロックなど、一定の周期を繰り返す波形を生成します。

M NewSig:/filtref/clkx2 <patterr< th=""><th>i : clock></th><th>×</th><th></th><th></th></patterr<>	i : clock>	×		
Specify the Clock Pattern	Clock Attributes Initial Value		Clock Attributes	
			Initial Value	初期値
	5 In	ns 💌	Clock Period/Time Unit	クロック周期/単位
	Duty Cycle 50		Duty Cycle	デューティ比
1				
<u>.</u>	< Previous Finish	Cancel		





Constant

入力した値の一定の波形を生成します。

▶ NewSig:/filtref/d <7:0> <pa< th=""><th>ttern : constant> 🛛 🛛 🛛</th><th></th><th></th><th></th></pa<>	ttern : constant> 🛛 🛛 🛛			
Specify the Constant		Const	ant Attrib	outes
Pattern Attributes.	Constant Attributes	Value		値
	Value			
	< Previous Finish Cancel			

Random ۵

> ランダムタイプや初期値、シード値を入力して、シード値によって異なるパターンの波形を生成します。 シード値には整数を入力してください。

NewSig:/filtref/d <7:0> <pat< th=""><th>tern : random></th><th>X</th></pat<>	tern : random>	X
Specify the Random Pattern Attributes.	Random Attributes Initial Value O	
	Pattern Period	Time Unit
	Random Type Uniform	Seed Value
1	Uniform Normal Exponential	; Finish Cancel
	Poisson	

Random Attributes				
Initial Value	初期値			
Pattern Period/Time Unit	パターン周期/単位			
Random Type	ランダムタイプ			
	Uniform	一様乱数		
	Normal	正規乱数		
	Exponential	指数乱数		
	Poisson	ポアソン乱数		
Seed Value	シード値(デフォル	トは 5)		





初期値と、繰り返しのパターンを入力します。また、そのパターンの繰り返し回数を入力して、一定パターンの波形を生成します。

NewSig:/filtref/d <7:0> <patte< th=""><th>rn : repeater> 🛛 🛛</th></patte<>	rn : repeater> 🛛 🛛
NewSig:/filtref/d <7:0> <patter< b=""> Specify the Repeater Pattern Attributes.</patter<>	Repeater Attributes Initial Value 00000000 Change Period Time Unit 100 Values 00000011
	00000111 More Fewer
	C Never C Count Times 12
	< Previous Finish Cancel

Repeater Attributes				
Initial Value	初期値			
Change Period/Time Unit	変化周期/単位			
Values	値 (More/Fewer ボタ	ンで追加削除)		
Repeat	リピート回数			
	Forever	前画面で指定した時 間分を繰り返す。		
	Never	1回のみ。		
	Count	Times 欄で指定した 回数分を繰り返す。		



カウント信号の最初と最後の値、時間周期、ステップ数、繰り返し回数を入力します。また、カウント方向 とカウントの種類を選択して、カウント波形を生成します。

M NewSig:/filtref/d <7:0> <pat< th=""><th>tern : counter></th></pat<>	tern : counter>
NewSig:/filtref/d <7:0> <pat< u=""> Specify the Counter Pattern Attributes.</pat<>	tern : counter> Counter Attributes Start Value End Value 00000000 01000000 Time Period Time Unit 50 ns Counter Type Count Direction Binary Up Step Count 1
	Repeat Forever Never Count Times
	< Previous Finish Cancel

Counter Attributes					
Start Value/End Value	開始値/終了値				
Time Period/Time Unit	周期/単位				
Counter Type	カウンタのタイプ				
Count Direction	カウンタの増減パターン				
Step Count	増減値				
Repeat	リピート回数				
	Forever	前画面で指定した時 間分を繰り返す。			
	Never	1回のみ。			
	Count	Times 欄で指定した 回数分を繰り返す。			

作成した波形パターンを修正する場合は、Wave ウィンドウ内で目的の信号を右クリックにて選択し、 Create/Modify Waveform を選択します。起動した Create Pattern Wizard を用いて作成と同様の作業を実 行し波形を修正してください。

ELSEN



2-5-3. 波形の編集

Wave ウィンドウにおいて、波形を修正できます。

- ① Edit Mode アイコン <u>北</u>を選択し、編集モードにします。
- ② ツールバーの Wave Edit を表示させます。



③ Wave ウィンドウ内で編集する信号の範囲を選択します。

💼 Wave 🚃									_		
∲ ⊒•		Msgs									
Cdit:/p	oush_entup/elk oush_entup/aelr	1	<u> </u>				ЛЛ		Ш	<u> </u>	M
4₽0	Now	0 ps) DS	1	1	1	· ·	I	1	1 4000	і і 00 ps
ê 🖉 😑 📃	Cursor 1	51942 ps	5194	2 ps	1446'	96 ps:					
💼 🧨 😑	Edit Cursor	196638 ps				1966:	38 ps				

④ 編集する内容に合わせて、編集パターンの Wave Edit コマンドを選択し、表示したウィンドウ内の値を 入力した後 OK ボタンをクリックします。



⑤ 入力内容に伴った波形が生成されます。



Wave Edit コマンド	
Insert Pulse	カーソルがアクティブな部分にパルスを入力します。
Delete Edge	カーソルがアクティブな部分のエッジを消去します。
Invert	選択した範囲の値を反転させます。
Mirror	選択した範囲の波形を反転させます。
Change Value	選択した範囲の値を、指定した値に変更します。
➡ Stretch Edge	エッジを前方か後方に移動させます。なお、他のエッジも前方、後方に移動します。
Move Edge	他のエッジを変更させずに、エッジを前方か後方に移動させます。
Extend All Waves	作成されたすべての波形を、記入した時間の値だけ延長させます。なお、延長 後にコマンドを取り消すことはできません。

2-6. テストベンチの生成

作成した入力波形を HDL 言語に変換し、テストベンチを生成します。

- ① 波形が入力された Wave ウィンドウを選択し、アクティブにします。
- ② File メニュー \Rightarrow Export \Rightarrow Waveform... を選択します。



③ Export Waveform ウィンドウ内の各項目を指定し、OK ボタンをクリックします。

M Export Waveform			×
Save As C Force File C EVCD File • V	HDL Testbend	h 🔿 Verilog) Testbench
Start Time	End Time		Time Unit
0	1000		ns 🔻
Design Unit Name			
filtref			
-File Name /work/qdesigns/fir_filter/fu	nc_sim/filtr	ef_tb.vhd	Browse
		ОК	Cancel



ModelSim/ModelSim-Altera 入力波形(テストベンチ)作成機能を用いたシミュレーション



Save As	ファイル形式を選択します。 VHDL Testbench : VHDL のテストベンチ Verilog Testbench : VerilogHDL のテストベンチ
Start Time/End Time/Time Unit	生成する範囲を指定します。
File Name	保存フォルダを指定し、ファイル名を入力します。

④ Export Waveform ウィンドウで入力した値に伴った波形が、選択した形式で生成されます。

Ln#	
1	LIBRARY ieee ;
2	LIBRARY std ;
3	USE ieee.std_logic_1164.all ;
4	USE ieee.std_logic_textio.all ;
5	USE ieee.std_logic_unsigned.all ;
6	USE std.textio.all ;
7	ENTITY filtref_tb IS
8	END ;
9	
10	ARCHITECTURE filtref_tb_arch OF filtref_tb IS
11	SIGNAL d : std_logic_vector (7 downto 0) ;
12	SIGNAL clkx2 : STD_LOGIC ;
13	SIGNAL yvalid : STD_LOGIC ;
14	SIGNAL yn_out : std_logic_vector (7 downto 0) ;
15	SIGNAL newt : STD_LOGIC ;
16	SIGNAL clk : STD_LOGIC ;
17	SIGNAL follow : STD_LOGIC ;
18	SIGNAL reset : STD_LOGIC ;
19	COMPONENT filtref
20	PORT (
21	d : in std_logic_vector (7 downto 0) ;
Wave :	🛛 🗟 filtref_tb.vhd 🗙

⑤ Wave ウィンドウを閉じます。

2-7. テストベンチの追加登録とコンパイル

エクスポートしたテストベンチをプロジェクトに追加登録し、コンパイルします。

2-7-1. **テストベンチの追加**

- ① Project ウィンドウを選択して、Project メニュー ⇒ Add to Project ⇒ Existing File... を選択します。
- Add file to Project ウィンドウの Browse メニューよりエクスポートしたテストベンチを選択した後 OK ボタンをクリックします。
- ③ Project ウィンドウに指定したテストベンチが追加されたことを確認します。

2-7-2. テストベンチのコンパイル

- ① Project ウィンドウにおいて、登録したテストベンチを選択します。
- ② 右クリックより Compile ⇒ Compile Selected を選択して、テストベンチをコンパイルします。



2-8. デザインのロード

作成したテストベンチをシミュレーションの最上位デザインとしてロードを実行します。(テストベンチが追加 されたことで、プロジェクトの最上位デザインはテストベンチになりました。)

- ① Simulate $\prec = \neg \rightarrow$ Start Simulation を選択します。
- ② 必要に応じ Libraries タブにてライブラリを登録します。
- ③ Design タブより、現プロジェクトのワーキング・ライブラリを展開し、最上位デザイン(テストベンチ)を選択し、OK ボタンをクリックします。

M Start Simulation			
Design VHDL Verilog Librarie	s SDF Othe	rs	<u>«</u> »
▼ Name	Туре	Path	
work	Library	work	
	Entity	F:¥design_sample¥work¥qdesign	s¥fir_filter¥a
🕂 🔁 accum	Entity	F:¥design_sample¥work¥qdesign	s¥fir_filter¥a
	Entity	F:¥design_sample¥work¥qdesign	s¥fir_filter¥f
– E filtref_tb	Entity	F:/design_sample/work/qdesigns,	/fir_filter/fur
A filtref_tb_arch	Architectu	ure	
🕂 🔁 hvalues	Entity	F:/design_sample/work/qdesigns,	/fir_filter/hv
🔁 🔁 mult	Entity	F:¥design_sample¥work¥qdesign	s¥fir_filter¥r
🔁 🔁 state_m	Entity	F:¥design_sample¥work¥qdesign	s¥fir_filter¥s
			••••••••••••••••••••••••••••••••••••••
Design Unit(s)		Resolu	ition
work.filtref_tb(filtref_tb_a	rch)	defa	ult 🚽
Optimization			
Enable optimization		Optimizat	ion Options
		[OK Cancel

④ デザインのロードが実行され、sim ウィンドウが追加されます。





ELSENA

2-9. シミュレーションの実行

- シミュレーションでモニタする信号を Wave ウィンドウに登録します。
- sim ウィンドウのテストベンチを右クリックで選択し、Add Wave を選択します。信号がと登録された状態の Wave ウィンドウが起動します。

💂 sim - Default 🚃 🗰 🖬 🖾 🔺 Objects 🔤 📰 🖆 🗹 🗶 📊 🖬 🗶 🗾 🖬 🖉 🖉								
Instance	🛆 Desiç 📩 💌 Name		Value	Mode		🔁 🕶	Msgs	
- filtref_tb	View Declaration			Interr	Π	+		
	View Instantiation		U 	Interr		/filtref_tb/clkx2	U	
→ → acc_inst	Add Ways		U 			<pre>/filtref_tb/yvalid</pre>	U	
	Add Wave New		U	Interr		+		
	Add Wave To		U	Interr	H	/filtref_tb/newt	U 	
line120	Add Dataflow Ctrl	+D	U	Interr	H		0 11	
line131	Add to	· •	U	Interr	Ľ	<pre>/filtref_tb/reset</pre>	U	
🔁 🗾 mult_inst								
	Copy Ctrl Find Ctrl	+C			Ш			

下位階層の内部信号をモニタする場合には、Objects ウィンドウ(View メニュー \Rightarrow Objects)を表示させます。sim ウィン ドウの下位デザインを指定し、Objects ウィンドウに表示された内部信号をドラッグ&ドロップにて Wave ウィンドウへ登録してください。

- ② メイン・ウィンドウ内の Run Length 部分に実行するシミュレーションの時間(数字と単位)を入力します。
- ③ Run ボタンをクリックすると、Run Length に入力した時間だけシミュレーションが実行されます。



この他に、Transcript ウィンドウを活用しコマンド実行する方法があります。シミュレーション実行は run コマンドを使用します。時間/単位を入力後、Enter キーにて実行します。

例: VSIM> run 100ns

④ Wave ウィンドウにシミュレーション結果が波形表示されます。出力結果を確認します。

🔢 Wave - Default						
💫 -	Msgs					
<pre>/filtref_tb/d /filtref_tb/clkx2 /filtref_tb/yvalid /filtref_tb/yn_out /filtref_tb/yn_out (7) (6) (5) (4)</pre>	00010000 0 0 00000000 0 0 0 0 0 0 0	00010000				
 → (3) → (2) → (1) → (filtref_tb/newt → /filtref_tb/clk → /filtref_tb/follow 	0 0 0 0 1 1	Radix Format Combine Signals Group Ungroup	Global Signal P Symbolic Binary Octal Decimal	Radix 信号名で 右クリッ 基数を変	を選択(ハ ク ⇒ Ra 変更できま	イライト)し、 dix によって す。
/filtref_tb/reset	0	Force NoForce Clock	Unsigned Hexadecimal ASCII Time Eixed Point			
Now	1000000 p0 ps		 Use Global Ra Numeric Enum Symbolic Enun 	dix s ns	8000	00 ps



2-10. シミュレーションとプロジェクトの終了

シミュレーションとプロジェクトを終了します。

- ① Simulate $\forall = a \Rightarrow$ End Simulation でシミュレーションを終了します。
- ② Project ウィンドウを選択し、File メニュー ⇒ Close Project でプロジェクトを終了します。

以上で、入力波形(テストベンチ)作成機能を用いたシミュレーションは終了です。





<u> 改版履歴</u>

Revision	年月	概要
1	2013年2月	新規作成





免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご 3. 一報いただければ幸いです。

株式会社アルティマ : 〒222-8563 横浜市港北区新横浜 1-5-5 マクニカ第ニビル TEL: 045-476-2155 HP: <u>http://www.altima.co.jp</u> 技術情報サイト EDISON : <u>https://www.altima.jp/members/index.cfm</u>

- 株式会社エルセナ : 〒163-0928 東京都新宿区西新宿 2-3-1 新宿モノリス 28F TEL: 03-3345-6205 HP: <u>http://www.elsena.co.jp</u> 技術情報サイト ETS : <u>https://www.elsena.co.jp/elspear/members/index.cfm</u>
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 本資料は製品を利用する際の補助的な資料です。製品をご使用になる場合は、英語版の資料もあわせてご利用ください。 5.